Vehicle Classification Using Neural Networks and Image Processing

K. W. Ong[1], S. L. Loh[1*], T. H. Cheong[2]
[1]Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
[2]Faculty of Education, Universiti Teknologi MARA Cawangan Selangor, Kampus Puncak Alam, 42300 Bandar Puncak Alam, Malaysia
[*]corresponding author's email: slloh@utem.edu.my

Abstract – *Vehicle classification is getting important especially in security systems, surveillance, transportation congestion reduction, and accident prevention. However, it is difficult to classify the traffic objects due to the poor quality of images from videos. Hence, image processing techniques are applied to increase the accuracy of the result. YOLO v5 and Faster R-CNN are two popular methods in object detection. Hence, these two algorithms are being chosen to compare the performance of the proposed vehicle classification scheme. The aim of this study is to propose a vehicle classification scheme where YOLO v5 algorithm and Faster R-CNN algorithm are being implemented separately into vehicle classification, followed by comparison in terms of performance. In this study, vehicles are being classified into five classes, namely motorcycle, car, van, bus and lorry. The labeled dataset is being split into training set and validation set and then trained under algorithm YOLO v5 and Faster R-CNN separately. Experimental results show that YOLO v5 performs better with the mean average Precision, Precision, and Recall rate up to 0.91, 0.81, and 0.86, respectively.*

Keywords: *Faster R-CNN algorithm, Neural network training, Vehicle classification, YOLO v5 algorithm*

Article History
Received 12 May 2022
Received in revised form 1 September 2022
Accepted 30 September 2022

I. Introduction

Vehicle classification systems play an important part in the development of intelligent transportation systems, given the exponential growth of vehicle production around the world. With the increase in vehicle traffic, problems such as traffic accidents, congestion, and air pollution caused by traffic have recently appeared. Among them, traffic accidents are a particularly difficult issue to solve. In criminal investigations of traffic accidents, it is essential to have automated technology to search for suspicious cars.

By using a vehicle classification system, it able to solve this problem and quickly search for suspicious vehicles. Besides, it is crucial in a variety of industries, including minimizing traffic accidents, charging tolls, avoiding traffic congestion, monitoring terrorist activities, security, and surveillance systems and so on [1].

The social economy's fast expansion has a substantial influence on many elements of society, such as transportation. Increased vehicle numbers result in major concerns such as accidents, auto robberies and traffic congestion. Vehicle classification is vital and capable of solving these problems in order to deal with these challenges. In recent years, vehicle classification has been a more popular topic of research in the field of vehicle classification. Vehicle categorization is a valuable approach for toll plazas, traffic monitoring, avoid traffic jams and accidents, and terrorist activity detection, among other transportation systems.

There are various types of vehicle classification have been proposed and different methods used in vehicle classification will result in different accuracy rates. The accuracy of vehicle classification depends on which method is used to classify the vehicle. For example, some vehicle classifications can only be detected and classified by the front and rear points of the vehicle, which will result in some vehicles not being detected and classified. Therefore, some techniques should be

This is an Open Access article distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License, permitting copy and redistribution of the material and adaptation for commercial and uncommercial use.

used to detect the multi-viewpoint of the vehicle. In this paper, multi-viewpoint of vehicles is used as input images for training purposes. Different data annotation and data augmentation techniques are applied to produce a richer databased based on the acquired images.

II. Literature Review

There are three main parts in the literature review as follows.

A. Vehicle Classification Techniques

There are generally two commonly used vehicle classification techniques, namely hardware-based vehicle classification and software-based vehicle classification.

Vehicle classification using hardware requires the assembly of some tools or equipment for vehicle classification. This is a traditional method of vehicle classification. It has low visibility and provides less information compared to vehicle classification using software.

Vehicle classification using software is a video-based approach for detecting and classifying vehicles. Vehicle classification using software is more popular than vehicle classification that using hardware because the installation of vehicle classification that using software is easier compared to the vehicle classification that using hardware.

Though software-based vehicle classifications have some drawbacks, for example poor processing power and a real-time capability that is limited, they are still superior to hardware-based classifications. Software-based vehicle classification is more favourable due to its ease of installation compared to other. Besides, in terms of maintenance, it is also easier to be maintained than the others. In other aspects, for example multilane recognition, and availability of vehicle synchronization information is also dominant. For example, Xiaoxu Ma et al. [2] showed that geometrical parameters such as area, breadth, aspect ratio, and rectangularity are used to distinguish between different size groups of vehicles.

Image processing is a technology for improving original image. The original or raw image can be received from camera or photos that obtained in daily life. The image is a graphic item with a rectangular shape. Sharpening, brightness improvement, blurring and edge improvement are examples of image improvement techniques, fall within the category of image processing. Image processing include concerns such as image description, compression methods, and a wide range of complicated operations that may be performed on image data.

Detecting vehicles is the initial stage in vehicle classification. Vehicle detection involves motion expression, tracking, and behavior analysis, all of which serve as the foundation for subsequent processing to gain a high classification success rate.

Vehicle detection can be divided into two categories which are appearance-based and motion-based. For an appearance-based approach, factors such as a vehicle's texture, color, and shape are considered [3]. In a motion-based approach, the movement properties are employed to distinguish the vehicles from the static background scenes [4].

B. Image Classification Techniques

In the field of image classification, there are numerous deep learning models, each with its unique set of strengths. Convolutional Neural Network (CNN) models were put to the test. Convolutional Layer, Pooling Layer, and Fully Connected Layer are the three main components of CNN. Fig. 1 shown the overview of CNN architecture.

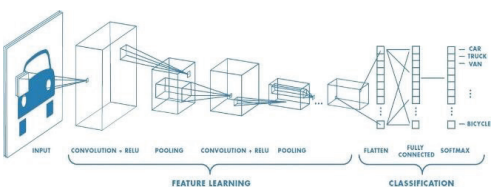


Fig. 1. Overview of CNN architecture [5]

Filters and feature maps are included in the Convolutional Layer. Filters are processors that work on a certain layer. These filters are not interchangeable. After that, they need to generate feature maps. To generate a feature map, it is necessary to input pixel values. One filter layer's output is a feature map. Filter is applied to the entire image, one pixel at a time. A feature map is created by the activation of a few neurons. To reduce dimensionality, a pooling layer is used. To generalize characteristics learned from prior feature maps, it requires to add the pooling layers when the convolutional layers is complete. This reduces the risk of overfitting throughout the training process. Fully connected layer is finally employed to assign features to likelihoods after collecting and merging features from convolutional layers and pooling. Linear activation functions or softmax activation functions are used in these layers.

One of the four fundamental problems in computer vision is object detection, which serves as the foundation for tasks like instance segmentation and object tracking. Deep learning's capabilities are primarily used in recent

object detection. Target detection can also be used in a variety of applications, including pedestrian detection, facial recognition, and text recognition. For example, in a photograph, the object is framed first, and then the object's identity is determined, a process known as object localization and image classification [6]. Object detection is the process of locating and framing all items in an image that you believe belong to the same class as known objects, as well as assigning class names and reliability scores. You Only Look Once (YOLO), Single Shot Detector (SSD), and Region-based Convolutional Neural Networks (RCNN) are three popular object detection algorithms.

The YOLO method employs a distinct CNN model to accomplish end-to-end target identification. The complete system is the input picture is enlarged to 448x448, then delivered to the CNN network, and lastly the network prediction results are processed to yield the identified target. YOLO's training approach is similarly end-to-end [7].

The SSD network's main design concept is to extract features in a hierarchical order, then perform boundary regression and classification in that order. Low-level feature maps can represent low-level semantic information, which can improve the quality of semantic segmentation and is suitable for small-scale target learning, because feature maps at different levels can represent semantic information at different levels. High-level feature maps are useful for deep learning of large-scale objects because they can represent high-level semantic information and smooth segmentation results [8].

The R-CNN model initially chooses various suggested areas from the picture (e.g., anchor boxes are a selection method), and then labels their categories and bounding boxes (e.g., offsets). They then utilized a CNN for forward computation to extract characteristics from each suggested area [9]. Then, the attribute of each region is applied to forecast their classes and bounding boxes. Specifically, R-CNN comprises of four essential elements. First, a selective search is performed on the input picture to choose numerous high-quality recommended areas [10].

C. Related Works

Pyong-Kun Kim and Kil-Taek Lim [11] proposed four concepts for improving performance on images with varied resolutions. The Deep Learning approach is critical for multi-viewpoint images, while the bagging method makes the system more resilient, data augmentation aids classification, and post-processing compensates for data imbalances. They integrate these approaches to create a new framework for categorizing vehicle types. A vehicle type classification system constructed by combining these approaches performed well with accuracy 97.84%. However, this combination requires a long time to train a deep learning and requires high training sample quantity.

Fukai Zhang and Ce Li [12] in their research presented a vehicle detection framework that improves on the Single Shot MultiBox Detector (SSD), this is capable to detect numerous types of vehicles in real time. The DP-SSD strategy offers the employment of distinct feature extractors for localization and classification tasks in a same network, as well as deconvolution (D) and pooling (P) between layers in the feature pyramid to improve these two feature extractors. Extensive experiment on datasets demonstrate that their method exceeds several existing algorithms in terms of accuracy and can recognize objects in real time. Anyway, the batch normalization layer was deleted owing to GPU memory constraints, and hence reduce the speed of network training and reduce model accuracy to 77.94%.

Ziwen Chen, Lijie Cao and Qihua Wang [13] presented a vehicle recognition approach based on high-resolution photographs recorded by UAVs, which addresses that existing object detection methods are constrained by images and object size. High-resolution photos might hinder the speed of the network for identifying tiny tar particles. So, the author adopted the YOLO v5 object detection method as the baseline. They evaluated the precision of their algorithm using Precision, Recall, and mAP and they obtained accuracy up to 91.9%, 82.5% and 89.6%, respectively. With such a great result, the operating speed of their algorithm is the main goal that they need to improve in the future.

In 2017, a deep-learning-based segment-before-detect method for segmenting [14], then detecting and classifying several types of vehicles such as car, van, lorry in high-resolution photos. With this, inspection on large datasets with visually comparable classes is possible, and object recognition and classification are also possible, as well as highlight the utility of a subclass modeling technique. To conduct vehicle classification, the author trained a Convolutional Neural Network (CNN). Outstanding semantic mapping results are generated by deep fully convolutional networks with accuracy of 67%-80%. This shows convolutional neural network works efficiently in vehicle classifications.

Recently, an improved visual background extractor was introduced [15] to detect illegal parking vehicles. This method develops the ViBe approach to detect illegal parking by modifying the background model update mechanism with the static region extraction and vehicle verification processes and integrating tracking processes. Experimental results show that the values of precision, recall, and f-measure are 100%, 60%, and

75% respectively. The accuracy can be further improved since it managed to overcome all false-positive problems. Table I shows the comparison between these classification methods.

From the previous related works, it can be concluded that many researchers applied neural network in vehicle classification, and they obtained outperformed results from their proposed vehicle classification systems, but there is so far no research in comparison between YOLO v5 and Faster R-CNN in vehicle classification. Therefore, YOLO v5 algorithm and Faster R-CNN algorithm will be implemented in this project for vehicle classification and their performance and results will be analysed.

TABLE I
COMPARISON BETWEEN VARIOUS CLASSIFICATION METHODS

Method	Features Used	Accuracy
Bagging and Convolutional Neural Network [11]	Deep learning, Bagging, Data augmentation, Postprocessing	97.84%
Deconvolution and Pooling Single Shot MultiBox Detector (DP-SSD) [12]	Feature Concatenation, Default Box with Feature Concatenation, Deep Nets Training	77.94%
YOLO v5 [13]	ReLU and Sigmoid activation function	91.9%
Convolution Neural Networks [14]	SegNet for Semantic Segmentation, Small Object Detection, CNN-Based Vehicle Classification	67%-80%
Faster R-CNN [9]	Bounding box classification	67.2%

III. Methodology

The dataset used for neural network training consists of different vehicle photos. The vehicle photos were taken using a smartphone camera with RGB color space, and store in JPG format. All the photographs were taken in real daylight conditions. Each of the photos are labelled into different classes of vehicle, namely motorcycle, car, van, bus and lorry. The dataset is labelled using Roboflow. This section will go into greater detail on the project implementation process, which is depicted in a flowchart. To achieve the objectives of this project, the flow chart of the project is shown in Fig. 2.

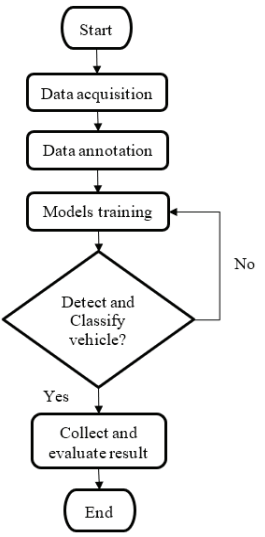


Fig. 2. Flow chart of vehicle classification

To begin, data acquisition can be completed by collecting the image dataset of vehicles. Next, the vehicle images are labelled according to their respective classes, namely motorcycle, car, van, bus and lorry. The label work is done by using Roboflow annotation tool. Then, labeled dataset is being split into training set and validation set, and being used to train different models to classify vehicles. After model training, result is collected, and training performance is analyzed.

A. Data Acquisition

Data acquisition is the first thing to do to start the neural network training. The proposed system for vehicle classification is evaluated on a self-built dataset. The dataset collected for neural network training are images of vehicles. In order to have a reliable dataset, data was being collected on a shiny bright day to make sure the collected images for training use are in the best quality condition. Poor quality image may affect the performance of the trained model and directly reduce the efficiency of the proposed vehicle classification scheme.

The datasets that are used to train the neural network was filmed on the pedestrian bridge in Jalan Simpang Ampat, Penang at coordinate 5.272819, 100.476876 on 8th March 2022 at 3.00 pm. The bridge has the limit height of 5.4 m. The dataset is manually acquired from the pedestrian bridge using phone camera. The phone is mounted on the phone holder during shooting to reduce vibration. The setup platform is illustrated in the Fig. 3. The data was taken as images and videos, and some

image files are extracted from the video to be used for the neural network training. The total data acquisition is 255 images. The image consists of different vehicles, namely motorcycle, car, van, bus and lorry. Fig. 4 shows the example of images from the dataset and Fig. 5 shows an image from the video file.

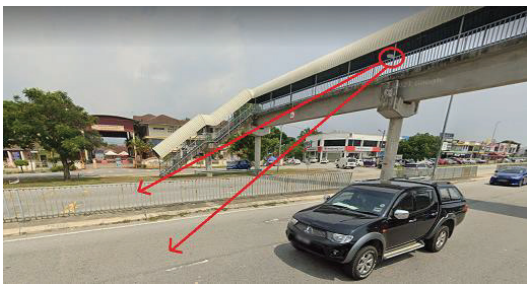


Fig. 3. Setup platform

Fig. 4. Images of the dataset

Fig. 5. Image of video

B. Data Annotation

After collecting sufficient image and video files from data acquisition, the next step is the data annotation. Data annotation is a process where Roboflow is used to label the image. Roboflow may execute pre-processing and augmentation procedures on the data. The action to get data into the correct format is known as pre-processing. Fig. 6 shows the image annotation. The image annotation was labelled with their respective name of object.

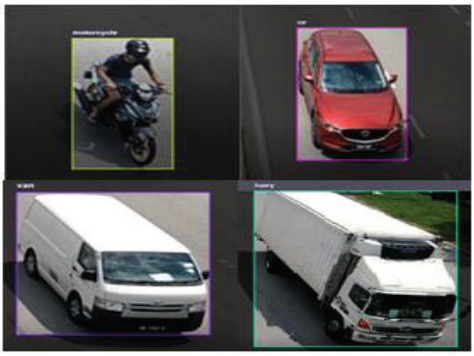
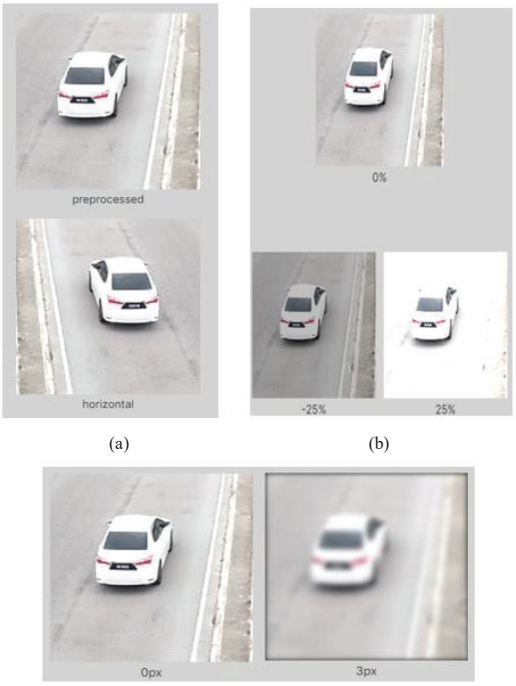


Fig. 6. Image annotation

After completing the label of image, the images will be separated into two groups; where 80% of the images are used as training set while another 20% as validation set. This percentage of allocation is a good start of models training [16]. Thus, from a total of 255 collected images, the set of training contains 204 images while the validation set contains 51 images. By applying image preprocessing to all the images from the dataset, this shortens up the training time and improves performance. All the images are auto oriented and resized to 416 x 416 pixels. Augmentation is a technique to increase the size of a dataset so that more data may be learned and trained. It improves model performance by reducing dataset memorization.

The geometrical operation as a data augmentation or data enhancement approach is proposed to boost the network's resilience to such shocks. By flipping a picture, flip augmentation increases model performance. Flip-augmentation, brightness-augmentation and blur-augmentation are the three augmentations used in the training. Flip-augmentation flips the image horizontally as shown in Fig. 7(a). The brightness of images is altered with -25% and +25% by bright-augmentation as shown in Fig. 7(b) to make the model be more resilient to lighting and camera settings changes. Blur-augmentation sets the blurriness of images to 3px as shown in Fig. 7(c). Gaussian blur is able to make model to be more resilient to camera focus. After augmentation, the size of dataset increases to 779 images, where the training set increased to 728 images to improve performance of the model.



(a) (b)

(c)

Fig. 7. Image augmentation (a) Flip, (b) Brightness and (c) Blur

C. Models Training

Models training is run using the Google Colab platform and python programming. The python code can be run through Google Colab. Google Colab provides free access to Google cloud computing capabilities for a variety of computing needs. Colab is a free-to-use cloud-based Jupyter notebook environment. Most importantly, there is no need to setup the workspace, and the notebooks one created may be updated by the user at any time. Google Colab provides free access to the "NVIDIA Tesla K80" GPU. Although it provides free GPU operation, the use of GPU is limited. Once the limit is exceeded, Google Colab cannot be used to run the program.

The YOLO v5 model is capable to detect little things in the background. Fig. 8 shows the architecture of YOLOv5 [17]. The YOLOv5 is a clone repository, and it installs the requirements to get started with YOLO v5. This will set the stage for object identification training and inference techniques in programming environment. The training environment is provided by Google Colab. YOLO v5 object detection data can be downloaded in own customized format. For a custom object detector, YOLOv5 is defined as the Model Configuration file. After the training, the validation metrics may be used to assess the performance of the YOLO v5 detector. Then the YOLOv5 training result is being evaluated and exported. Finally, YOLOv5 weight is saved for future use. The input images are resized to 416 × 416 pixels during the object detection process. The backbone, feature pyramid network, and detection head are the three fundamental structures of the YOLO v5 network model. The backbone network extracts feature from various images at various scales, the feature pyramid network is in charge of fusing features from various scales and passing them to the detection network, and the detection network is in charge of predicting the object category in the image and generating the object bounding box using the image features.

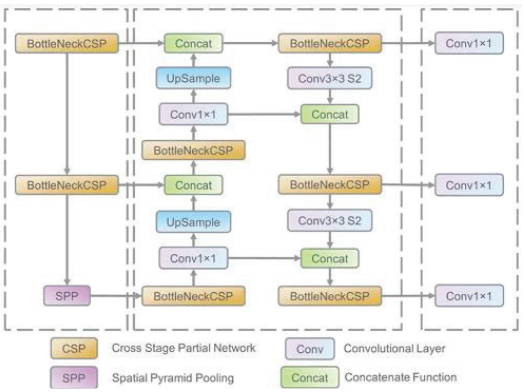


Fig. 8. YOLOv5's architecture

The design of faster R-CNN includes two primary modules: Region Proposal Network (RPN) and the Fast R-CNN object detector. To create a feature map, pictures passed through convolutional layers, then processed images of varying sizes (to avoid object shape deformation due to rescaling) are aggregated by RPN to anticipate a collection of objects with objectness scores (to judge whether it is an object). Fixed-length feature vectors are extracted from each proposal via RoI pooling. After that, the feature vectors are inserted into a layer sequence with two sibling outputs. The system generates a detection of the target item, and the other output creates four real values that reflect the bounding box position of the object. Faster R-CNN separates the detection framework into two steps, making it faster. The image is exposed to a feature extractor in the first step, known as RPN, and the highest feature map is utilized to anticipate bounding box recommendations. These ideas are then utilized to trim features from the top-most feature map, which are then supplied to Fast R-CNN for classification and bounding box regression in the second step. Although Quicker R-CNN is an order of

magnitude faster than Fast RCNN, it is constrained by the first stage CNN feature extraction and the second stage costly per-region calculation. The architecture of Faster R-CNN is shown in Fig. 9.

Image processing can be used to detect and classify different vehicles. It involves image acquisition, preprocessing, division, labeling, enhancement, and classification. A total of 255 images are collected and images are resized to 416 x 416 pixels. Data annotation is done using the Roboflow label tool by drawing a bounding box around the logo implemented in each image. Images are trained using the YOLOv5 object detection architecture and R-CNN architecture. Finally, the best trained model is used to run the simulation of vehicle detection and classification. The simulation is performed using the video file that have been captured.

IV. Results and Discussion

A. YOLO v5

The effectiveness of the network training can be verified using different metrics including mean Average Precision (mAP), Precision, and Recall. YOLO v5 algorithm and Faster R-CNN algorithm will be used in network training. The models are trained using Google Colab, which provides free access to powerful GPUs and requires no configuration. The total dataset used in network training is 728 images while 728 images were assigned to the training set and 51 images were assigned to the validation set. The training classes is divided into five classes namely motorcycle, car, van, bus and lorry.

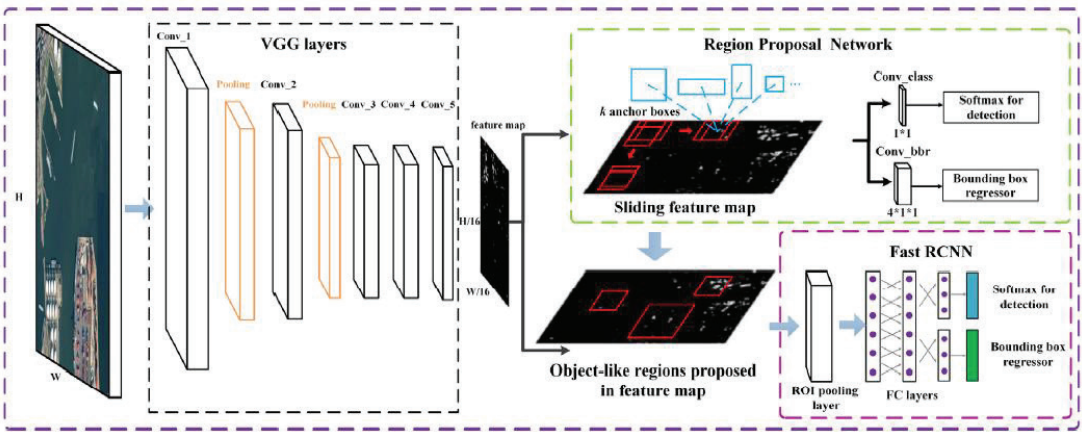


Fig. 9. Faster R-CNN's architecture

Training of YOLO v5 model for 100 epochs needs around 124 minutes. An epoch indicates the number of passes of the entire training dataset the machine learning algorithm has completed [6]. The mean Average Precision@.5 (mAP_0.5) obtained from YOLO v5 training is 0.912. The Precision and Recall obtained from YOLO v5 training is 0.809 and 0.859, respectively. Fig. 10 shows the mean Average Precision@.5, Precision and Recall. From the results, the model rapidly improves from 0 in mean Average Precision, Precision and Recall, then stabilizes after about 80 epochs.

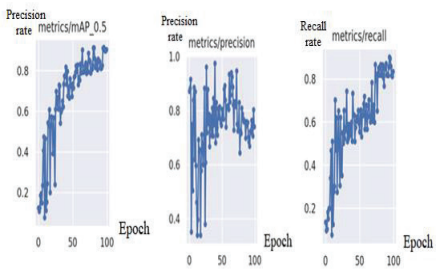


Fig. 10. Mean Average Precision@.5, Precision and Recall

Once the training has been completed, it will give the predictions from the valid dataset on the trained model. Fig. 11 shows the sample of predicted images for YOLO v5 algorithm. The name of the detected object and the probability of the detected object will be displayed on the top of bounding box. From Fig. 11, it shows that the

vehicle was detected correctly with high confidence rate. The proposed model presents impressive results where the different vehicles are detected with a confidence rate ranging from 0.7 to 0.9.

Fig. 11. Predicted images for YOLO v5

B. Faster R-CNN

On the other hand, the results gained from Faster R-CNN are shown in Fig. 12 to Fig. 14. The mean Average Precision@.5 (mAP_0.5) obtained from Faster R-CNN training is 0.8596. While the Precision and Recall obtained from Faster R-CNN training is 0.75 and 0.841, respectively. Fig. 12 shows the mean Average Precision@.5, Fig. 13 shows the Precision and Fig. 14 shows the Recall. From Fig. 13, the precision rate drops dramatically from 0.6958 at 1294 steps to 0.5458 at 1952 steps, and then increase steadily to 0.75 precision rate. The drop of the precision rate is due to the high learning rate. High learning rate will make the learning jumps over minima, and low learning rate can cause the process to get stuck. The learning rate is set to 0.0002 and it is standardized learning rate for Faster R-CNN training. The precision rate then increases steadily from 1952 steps to final steps.

Once the training has been completed, it will give the predictions from the valid dataset on the trained model. Fig. 15 shows the predicted image for Faster R-CNN algorithm. The name and the probability of the detected object will be displayed on the top of bounding box. The predicted image of vehicle using Faster R-CNN algorithm, as presented in Fig. 15, shows the vehicles are successfully being detected with high probability and with the correct name of object. This model presents impressive results where different vehicles are detected with probability ranging from 0.89 to 1. The other result of predicted images shows the vehicles are detected with probability 0.55 and 0.99, respectively, but the vehicles are not fully detected by the bounding box.

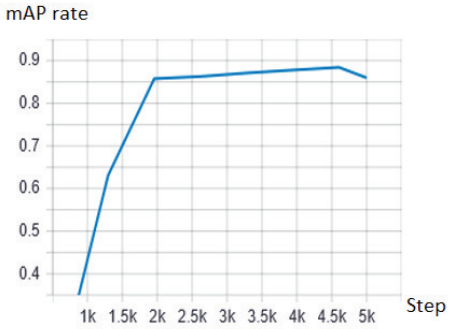


Fig. 12. Mean Average Precision@.5 from Faster R-CNN training

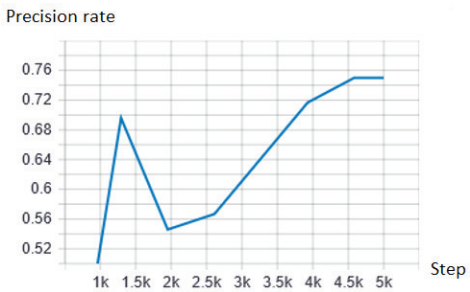


Fig. 13. Precision from Faster R-CNN training

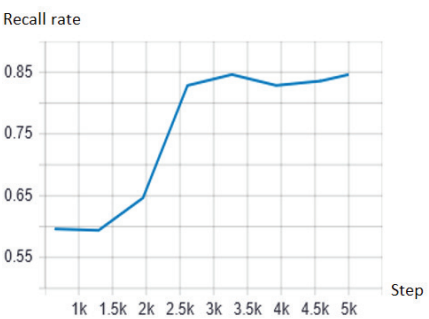


Fig. 14. Recall from Faster R-CNN training

Under the training with the Faster R-CNN algorithm, some of the results show that some parts of the vehicle are not detected. This is because the extracted feature maps are all single-layered and the resolution is relatively small. Therefore, this will affect the recognition of images with multiple scales and small objects.

Note that the number displayed on the bounding box represents the name of the detected object, as follows:

- 0 is bus
- 1 is car
- 2 is lorry

- 3 is motorcycle
- 4 is van

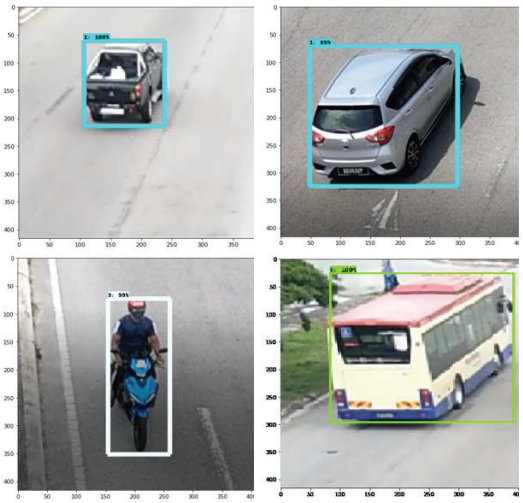


Fig. 15. Predicted images for Faster R-CNN

C. Performance Comparison of YOLO v5 and Faster R-CNN

Table II shows the comparison results of the two different algorithms which are YOLO v5 and Faster R-CNN trained using the same dataset. YOLO v5 presents higher mAP_0.5 compared to Faster R-CNN. The mAP is an evaluation metric used to evaluate network performance in the field of object detection. mAP@0.5 is the area under the P-R curve of the network when the detection IOU ratio threshold is set to 0.5. The final mAP@0.5 achieved by the YOLO v5 algorithm is 0.912 compared to Faster R-CNN algorithm which is 0.869. Moreover, the Precision and Recall obtained by YOLO v5 are also better than Faster R-CNN. The Precision and Recall obtained by YOLO v5 are 0.809 and 0.788, respectively, while Precision and Recall obtained by Faster R-CNN are 0.788 and 0.835, respectively. A model with higher precision has lower false positive rate. A good model needs to have a high Recall since Recall is the ratio of accurately anticipated positive observations to all actual observations in a class.

From the results presented above, the YOLO v5 algorithm shows better performance while the Faster R-CNN algorithms reveals some disadvantages. YOLO v5 can clearly detect the vehicle with correct name of object with high confidence level, while Faster R-CNN can also detect the vehicle, but some vehicles are falsely detected. This is because the extracted feature maps for Faster R-CNN are all single-layered and the resolution is relatively small. Therefore, this will affect the recognition of images with multiple scales and small objects. Apart from having greater performance, YOLO v5 has a simpler design due to its one-stage detector, which combines the backbone and detector into one architecture and being trained concurrently.

TABLE II
RESULTS COMPARISON BETWEEN YOLO V5 AND FASTER R-CNN

Algorithm	mAP_0.5	Precision	Recall
YOLO v5	0.912	0.809	0.859
Faster R-CNN	0.8596	0.750	0.841

D. Simulation of Vehicle Detection and Classification

A video simulation is run using the best trained algorithm which is YOLO v5 algorithm. The video was filmed on the pedestrian bridge in Jalan Simpang Ampat, Penang at coordinate 5.272819, 100.476876. The time duration of the video is 39 seconds. In the video, the passing vehicles are detected and classified. The video simulation is done with speed of 0.5ms pre-process, 25.4 ms inferences and 0.8 ms Non Maximum Suppression (NMS) per image at shape. Fig. 16 shows the result of simulation of vehicle detection and classification. The video was uploaded to Youtube and can be achieved through the link; https://youtu.be/vzZ6Ro2fJ7s.

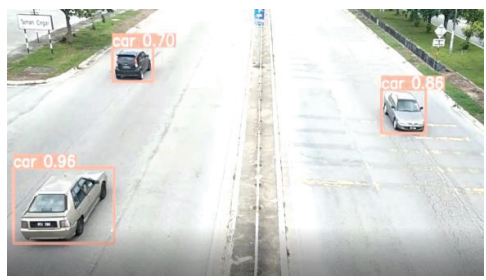


Fig. 16. Simulation of Vehicle Detection and Classification

A comparison was made using two different algorithms. By comparing the results from two algorithms, it is concluded that YOLO v5 performs better than Faster RCNN as it produces better results with higher mean Average Precision (mAP), Precision, and Recall. The trained YOLO v5 model is also used to run the video simulation of vehicle detection and classification and the passing vehicles in the video are successfully detected and being classified correctly. The faster RCNN creates a convolutional region of interest, whereas YOLO performs both detection and

classification. YOLO v5 is a better object detection algorithm since it is fully end-to-end training.

V. Conclusion

In this project, neural network is implemented in the vehicle classification scheme. Algorithms of YOLO v5 and Faster R-CNN have been tested to detect and classify vehicles and the performance is being compared in terms of mean Average Precision (mAP), Precision and Recall. From the experimental result, it can be concluded that the newly released model YOLO v5 performs better. The mean Average Precision@.5 (mAP_0.5), Precision and Recall for the YOLO v5 and Faster R-CNN are 0.912, 0.809, 0.859 and 0.8596, 0.75, 0.841, respectively. These results show that the YOLO v5 model is able to detect and classify the vehicles better than Faster R-CNN model.

In future, the size of dataset may be increased for models training to gain better performance. The images of the dataset can be taken with a professional camera to get clearer and higher pixel images. The clearer images allow the model to recognize objects better during training. Besides, the architecture of the models can be modified for better performance. For example, increase the number of filters in the architecture. The increased number of filters can increase the depth of the feature space, which helps to learn more levels of global abstraction.

Acknowledgements

Authors would like to express their gratitude to Research and Innovation Management Center (CRIM), and Universiti Teknikal Malaysia Melaka (UTeM) for the funding of this research.

References

[1] A. Iftikhar and A. Javed, "Video analytics algorithm for automatic vehicle classification (intelligent transport system)," *International Journal of Image, Graphics and Signal Processing*, vol. 5, no. 4, pp. 38–45, 2013.

[2] X. Ma and W. E. L. Grimson, "Edge-based rich representation for vehicle classification," in *Proc. of IEEE International Conference on Computer Vision,* vol. 1, Beijing, China, 2005, pp. 1185-1192.

[3] T. Gao, Z. G. Liu, W. C. Gao, and J. Zhang, "Moving vehicle tracking based on sift active particle choosing," in *Proc. of International Conference of Advances in Neuro-Information Processing*, Auckland, New Zealand, 2008, pp. 695–702.

[4] A. Ottlik and H.-H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner-city intersections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 211–225, 2007.

[5] Z. Kain, A. Y. Ouness, I. Sayad, S. Abdul-Nabi, and H. Kassem, "Detecting abnormal events in university areas," in *Proc. of International Conference on Computer and Applications*, Beirut, Lebanon, 2018, pp. 260-264.

[6] F. Gaillard, "Epoch (machine learning)| radiology reference article radiopaedia.org," Radiopaedia. [Online]. Available: https://radiopaedia.org/articles/epoch-machine-learning.

[7] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 7263–7271.

[8] J. Nelson, "YOLOv5 is here: state-of-the-art object detection at 140 FPS," Roboflow Blog, 10-Jun-2020. [Online]. Available: https://blog.roboflow.com/yolov5-is-here/.

[9] M. Zhou and X. Wang, "Object detection models of remote sensing images using deep neural networks with weakly supervised training method," *Scientia Sinica Informationis*, vol. 48, no. 8, pp. 1022–1034, 2018.

[10] T. Liu and T. Stathaki, "Faster R-CNN for robust pedestrian detection using semantic segmentation network," *Frontiers in Neurorobotics*, vol. 12, Issue October, p. 64, 2018.

[11] P. K. Kim and K. T. Lim. "Vehicle type classification using bagging and convolutional neural network on multi view surveillance image," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, 2017, pp. 914-919.

[12] F. Zhang, C. Li, and F. Yang, "Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation," *Sensors*, vol. 19, no. 3, p. 594, 2019.

[13] Z. Chen, L Cao and Q Wang, "YOLOv5-based vehicle detection method for high-resolution uav images," *Advanced Artificial Intelligence Technologies for Service Enhancement on the Internet of Medical Things*, Special Issue, pp. 1-11, 2022.

[14] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: vehicle detection and classification through semantic segmentation of aerial images," *Remote Sensing*, vol. 9, no. 4, p. 368, Apr. 2017.

[15] P. P. Yudha and Wahyono, "Improved visual background extractor for illegally parked vehicle detection," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 416- 426, 2022.

[16] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection", in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 936-944

[17] R. Xu, H. Lin, K. Lu, L. Cao and Y. Liu, "A Forest Fire Detection System Based on Assembling Learning", Forest, vol. 12, no. 2, p. 217,2021.